

```

//                      APPENDIX 1 Example C Source Code
//      File:
//          poff.cc
//      Program:
5 //      This program takes data for a pump-off controller. It looks for
//      transitions on the sensor line and reports them. This version
//      runs on a laptop; a microcontroller version comes next.
//      © 2003 Petrolects, LLC All rights reserved

#include <stdio.h>
10 #include <stdlib.h>
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>
#include <unistd.h>                //For usleep
15 #include "testtime.h"
#include "serial-bits.h"
#include "bin2a.h"
//      Function:      checkpulse
//      This function looks for pulses on the valve signal line, it measures
20 //      the duration of pulses and when a pulse is finished it writes the pulse's timestamp
//      and duration to a file.
//      States
//          The state machine in here has these states:
//          1      waiting for a pulse
25 //          2      pulse is active, waiting for it to end
//

```

```

//                                     APPENDIX 1 Example C Source Code
int checkpulse (double timeNow, bool valveSignal, ofstream& outStr)
{
    static int state = 1;
    5    static double onTime = 0.0;          // Time pulse begins
                                           // Run state machine with a switch statement

    switch (state)
    {
        // If the state is 1 (waiting), react if pulse is seen
    10    case (1):
            if (valveSignal == true)        // Transition detected
            {
                onTime = timeNow;           // Save pulse start time
                state = 2;                  // Go to pulse-active state
    15            }
            break;

            // If the state is 2 (pulse active), look for the end of pulse
        case (2):
            if (valveSignal == false)       // Transition detected
    20            {
                // Calculate and save the duration
                double duration = timeNow - onTime;
                outStr << onTime << "      "<<< duration << endl;
    25            cout << endl << onTime << "\t" << duration << endl;
                state = 1;                  //      Back to waiting state
            }
            break;
        default:
    30            cerr << "ERROR in checkpulse( ): Unknown state " << state << endl; return ( );
            };
        return state;
    }

                                           // © 2003 Petrolects, LLC All rights reserved
35

```

// APPENDIX 1 Example C Source Code

```
//-----  
//                               Main Program Body  
//-----
```

```
5  int main (int argc, char **argv)
    {
        double timeSec = 0.0;
        double duration = 0.0;           // Test duration in seconds
        const long us_per_step = 90000L; // Microseconds per test step
10  bool valveSig = false;              // Signal read from sensor
        C_testtimer theTimer;           // Create a timer object
        C_serialbits serPort (0x3F8);    // Serial port bits I/O object
                                         // Parse arguments with which we called

        if (argc != 3)
15        {
            cerr << "Usage: << argv[0] << " duration datafile " << endl;
            exit (-2);
        }
        if (sscanf (argv[1], "%lf", &duration) != 1)
20        {
            cerr << "Error: Cannot read test duration, I got " << duration << " sec " << endl;
            exit (-3);
        }
        ofstream outFile (argv[2]);
25  if (!outFile)
        {
            cerr << "Error: Cannot create file " << argv[2] << endl;
            exit (-4);
        }

30                                     // Set up serial port outputs, one true one false
        serPort.setDTR (true);
        serPort.setRTS (false);
        //      Do a bunch of tests
        while (timeSec < duration)
35        {
            timeSec = theTimer.now ( );
            valveSig = serPort.getDCD ( );
            cout << checkpulse (timeSec, valveSig, outFile)
                  << bin2a (serPort.getMSR ( )) << "
40                  << "      \r";

                                     //© 2003 Petrolects, LLC All rights reserved
```

```

//          APPENDIX 1 Example C Source Code
cout.flush ( );
cout << setprecision (3) << setw (6) << timeSec
    << " DCD=" << serPort.getDCD ( ) << "        \r";
5    cout.flush ( );

        // Wait for a hopefully fixed amount of time
if (usleep (us_per_step))
{
10    cerr << "usleep failed" << endl; exit (2);
}
}
//    outFile << index << "\t" << timeData[index]
cout << endl;
15 return 0;
}

        // © 2003 Petrolects, LLC All rights reserved

```